

Slides, videos, links and more:

<https://github.com/physicell-training/03-What-is-ABM>

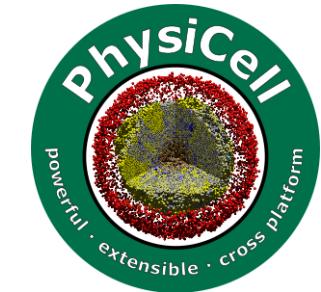
# Lesson 3: What is an agent-based model?

Paul Macklin, Ph.D.

 @MathCancer

## PhysiCell Project

last updated: February 11, 2020



# What is a discrete model?

- “**Discrete**” applies to discrete mathematics.
- **Continuum models** describe *continuous variables* with continuous (and differentiable) operations. The variables take continuous values. (e.g., positive real numbers)
  - **Example:** a cell population density  $\rho$  modeled with the Fisher's equation with diffusion ( $D$ ) and a birth rate ( $r$ ) up to a carrying capacity ( $\rho_{\max}$ ).

$$\frac{\partial \rho}{\partial t} = D \nabla^2 \rho + r \rho \left(1 - \frac{\rho}{\rho_{\max}}\right)$$

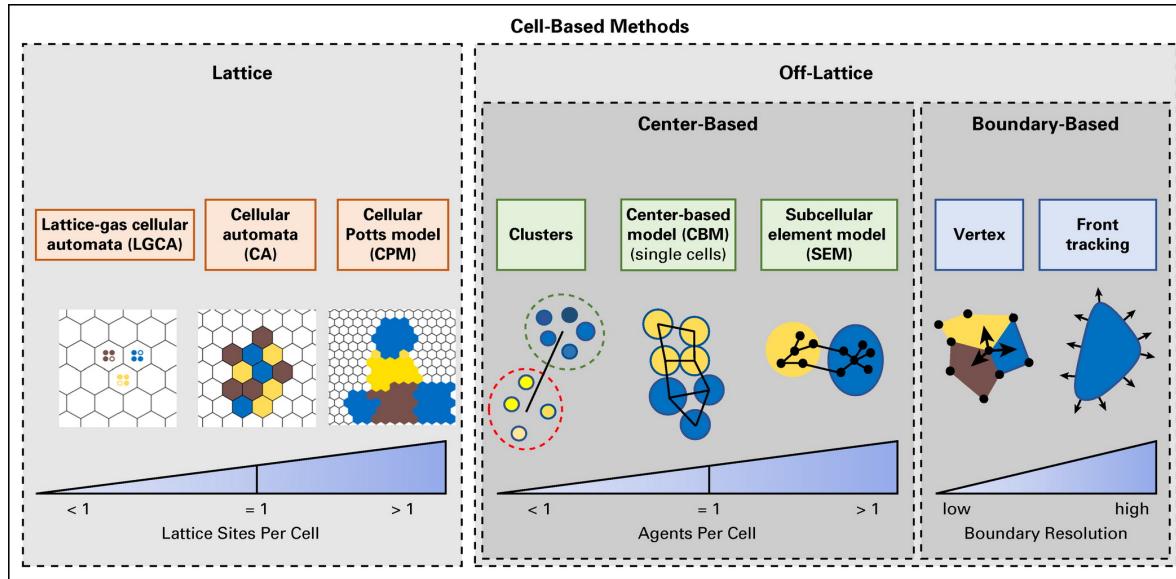
- **Discrete models** describe *distinct individuals* with discrete events. The variables tend to take discrete values. (e.g., Boolean or integer variables)
  - **Example:** A cell population  $X(t)$  models birth events as a Poisson process with rate  $r$ : Between now ( $t$ ) and the next time step ( $t + \Delta t$ ), there is a probability  $P = r\Delta t$  of a birth event that increases  $X$  by one.

# What is an agent-based model?

- An **agent-based model** (in biology) is a type of discrete model that simulates individual cells.
  - Also referred to as **individual-based models** or **cell-based models**.
- Agent-based models are often combined with continuum models of the microenvironment (e.g., partial differential equations for signaling factors), resulting in **hybrid discrete-continuum (HDC) models**.
- **Object-oriented programming (OOP)** is ideal for agent-based modeling:
  - Modeling work focuses on individual cells
  - Each cell is an independent *agent* that carries its own data, and has its own behavioral rules
  - **Use OOP:** Define a cell *class* with member data and methods. Each cell is an instance of that class.
- Agent-based models are a little closer to the biology:
  - Focus on modeling cells and their changing behavior.
  - Specific problems are then a matter of choosing the right rules.
  - You can tailor the level of detail: add molecular-scale biology to each cell if you need it.

# Main approaches

- Approaches can be classified as lattice or off-lattice, and by resolution.



Review: Metzcar et al. (2019). <http://dx.doi.org/10.1200/CCI.18.00069>

# Typical program flow

- Read parameters
- Set up microenvironment
  - Create meshes, initialize chemical substrates, diffusion solvers, etc.
- Set up cell agents
  - Define all cell types
  - Instantiate cells
- For each time:
  - Update microenvironment
    - ◆ Solve reaction-diffusion equations (as needed)
    - ◆ Solve tissue mechanics (as needed)
  - Update each cell's state
    - ◆ Sample environment
    - ◆ Run signaling model (as needed)
    - ◆ Update behavioral parameters based on signaling model and sampled environment
    - ◆ Run cell process models (growth, cycling, death, ...)
  - Calculate cell velocities
  - Update cell positions
  - Advance time

# Cellular Automata

## Approach:

- Divide space into a lattice
- **Spatial resolution:** Each lattice site holds 0 or 1 cell
- Update each lattice site based on rules

## Pros:

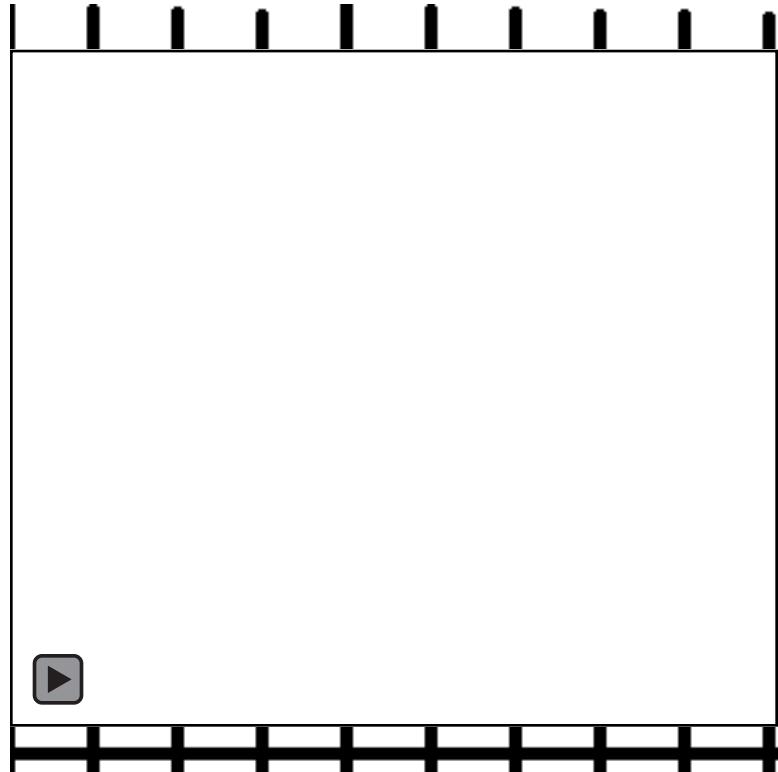
- Easy to program
- Very, very fast
- Easy to swap in new hypotheses

## Cons:

- Lattice effects (& hidden assumptions)
- All cells are the same size
- No tissue mechanics
- Update order biases
  - Usually solved with Monte Carlo (random update ordering)

## Ideal use case:

- Early qualitative tests of hypotheses



# Cellular Potts

## Approach:

- Use a smaller mesh to resolve cell morphology
- **Spatial resolution:** multiple pixels (or voxels) per cell
- Minimize a specially-chosen energy
  - Randomly try to swap pixels
  - Accept if energy is lower

## Pros:

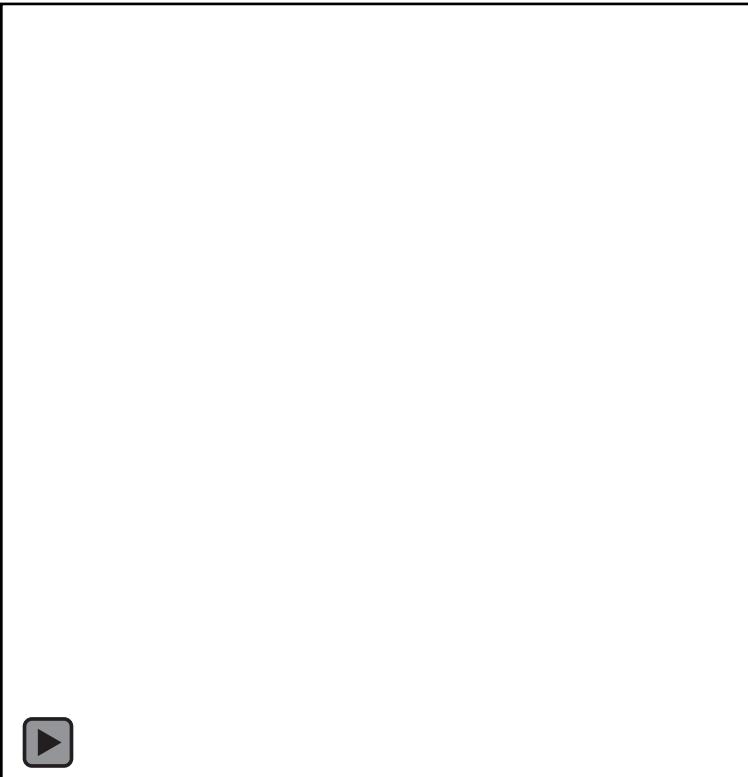
- Now we get cell shape and size
- More realistic
- Mature codes such as CompuCell3D and Morpheus

## Cons:

- No true time: Monte Carlo steps and “temperature”
- Have to translate biology into energy
- Unexpected correlations can pop up from the global energy
- Needs expert coding to be fast

## Ideal use case:

- Qualitative tests of hypotheses where mechanics matter



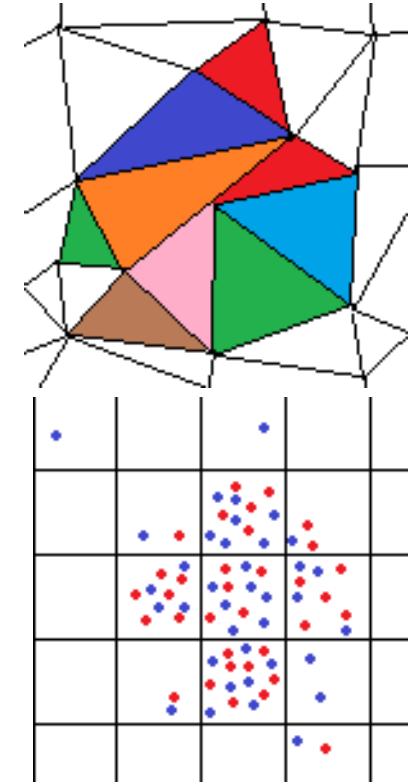
# Other lattice-based approaches

- Cellular automata on irregular meshes

- Use an irregular mesh
- Gets rid of grid bias issues
- But still no control over individual cell sizes
- Still no mechanics
- **Spatial resolution:** One pixel (voxel) per cell

- Lattice-gas

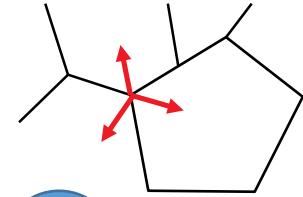
- Treat space as a series of connected boxes
- Each box contains one or more cells
- Pre-defined “channels” for cell movement between boxes
- A nice bridge towards continuum models
- **Spatial resolution:** < 1 pixel (voxel) per cell



# Key off-lattice approaches

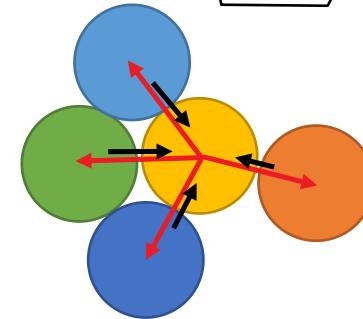
- **Vertex-based models:**

- Densely packed cells look like polyhedral
- Model the movement of the vertices, instead of the cells
- **Spatial resolution:** 1 polyhedron per cell, approximating morphology



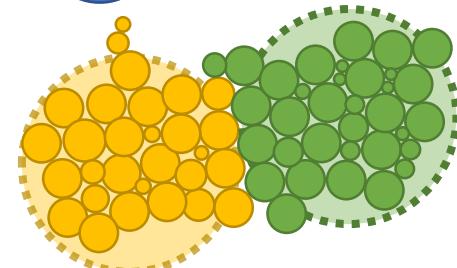
- **Center-based models:**

- Model movement of cell centers
- Write force balance laws for each cell (classic physics!)
- Update cell velocities and positions based on forces
- Append extra biology to each cell as needed
- **Spatial resolution:** 1 agent per cell



- **Subcellular element models:**

- Model movement of cell parts with individual agents (subcellular elements)
- Write force balance laws for each part (classic physics!)
- Update agent velocities and positions based on forces
- Model cell growth, cycling, death, etc. with additional operations:
  - ◆ grow, divide, shrink, fuse, and eliminate subcellular elements
- **Spatial resolution:** multiple agents per cell to approximate morphology



# Where does PhysiCell fit in?

- PhysiCell is an **off-lattice, center-based** modeling platform
  - **Spatial resolution:** one agent per cell
  - **A trick:** Use bigger agents to model cell collections or pieces of tissue.
- PhysiCell couples with PDE models of the microenvironment, making it a **hybrid discrete-continuum approach.**
  - Since most useful agent-based models are coupled to PDE models of the microenvironment, we simply refer to them as agent-based models.
- PhysiCell uses ODEs and other technical to model dynamical details in individual cells. This makes it **multiscale.**

# Next steps

**Super fast:** Please proceed to 4 (Introduction to PhysiCell)  
<https://github.com/physicell-training/04-PhysiCell-intro>

**Intermediate:** Please proceed to 4 (Introduction to PhysiCell)  
<https://github.com/physicell-training/04-PhysiCell-intro>

**Full training:** Please proceed to 4 (Introduction to PhysiCell)  
<https://github.com/physicell-training/04-PhysiCell-intro>

**More materials:** <https://github.com/physicell-training/master-list>

# Credits

**Lesson Planning:** Paul Macklin

**Slides:** Paul Macklin

**Recording:** Paul Macklin

**Post-production:** Paul Macklin, Drew Willis\*, Kali Konstantinopoulos\*

**Microapps:** Not applicable

\* denotes undergraduate researcher

## Funding:



## PhysiCell Development:

- Breast Cancer Research Foundation
- Jayne Koskinas Ted Giovanis Foundation for Health and Policy
- National Cancer Institute (U01CA232137)
- National Science Foundation (1720625)

## Training materials:

\* Administrative supplement to NCI U01CA232137 (Year 2)